

Mails zenden met Lazarus

Michaël Van Canneyt

May 13, 2012

Abstract

Er bestaan verschillende componenten om allerlei TCP/IP protocols af te handelen met Lazarus. E-Mail zenden (via het SMTP protocol) is hierop geen uitzondering. Een bijzonder gemakkelijke manier om E-Mails te verzenden is synapse.

1 Inleiding

Vaak is het nodig om programmatorisch een mailtje te sturen: een website die inschrijvingen afhandelt zal een bevestigingsmail sturen. Een windows service kan een fout melden aan de systeembeheerder, fo bijvoorbeeld waarschuwen als de harde schijf dreight vol te lopen.

Er bestaan verschillende klassebibliotheken die mails kunnen versturen. Indy of Synapse zijn er 2 van, die beide met Delphi en Lazarus kunnen gebruikt worden.

Synapse is een verzameling klassen die allerlei TCP/IP taken afhandelen. Synapse wordt niet standaard met lazarus meegeleverd, maar kan gedownload worden van:

<http://synapse.ararat.cz/>

Er is een `laz_synapse.lpk` package bestand. Het kan geopend worden met de lazarus package manager. Het volstaat het package 1 maal te compileren: het hoeft niet geïnstalleerd te worden in de IDE, want Synapse bevat geen componenten die op de component palette worden geïnstalleerd. Alle klassen moeten in code aangemaakt en gebruikt worden.

Voor zij die willen is er een apart package 'visualsynapse', met een verzameling componenten die op de component palette kunnen geïnstalleerd worden. Dit package wordt niet ondersteund door de maker van Synapse.

Eenmaal gecompileerd, kan het package als dependency van een project gekozen worden.

2 Mail via SMTP

Mail programmas maken gebruik van het protocol SMTP (Simple Mail Transfer Protocol) om mails te versturen. Dit wil zeggen dat ze het e-mail bericht afleveren aan de SMTP server (een MTA of Mail Transfer Agent), die de verdere aflevering op de juiste bestemming voor zijn rekening neemt. Doorgaans luistert een mailserver op TCP/IP poort 25.

Het SMTP protocol is vrij eenvoudig, en een mailtje zenden is dan ook een eenvoudige aangelegenheid. In Synapse kan dit met een functie die in de `smtplib` unit gedefinieerd is:

```
function SendTo(const MailFrom, MailTo,
```

```
Subject, SMTPHost: string;  
const MailData: TStrings): Boolean;
```

De 5 argumenten spreken voor zich:

MailFrom Het E-mail adres van de afzender.

MailTo De e-mail adressen van de ontvangers, gescheiden door commas. Er mogen namen in staan, dus iets als

```
Michael Van Canneyt <michael@freepascal.org>
```

is OK.

Subject Het onderwerp van de mail. Dit wordt aan de mail headers toegevoegd.

SMTPHost Het IP of DNS adres van de SMTP server die het bericht zal afleveren. Dit is meestal de uitgaande mail server van de internet provider.

MailData het e-mail bericht.

De `SendTo` functie maakt een instance aan van de klasse `TSMTPSend` en vult de nodige properties in, waarna de methode `MailTo` gebruikt wordt om naar alle opgegeven email adressen de mail te sturen. Als alles correct verlopen is, wordt `True` teruggegeven.

Sommige mailservers vereisen een gebruikersnaam/wachtwoord combinatie voor het versturen van mails. De functie `SendToEx` werkt identiek aan `SendTo`, maar staat toe een gebruikersnaam en wachtwoord op te geven:

```
function SendToEx(const MailFrom, MailTo,  
Subject, SMTPHost: string;  
const MailData: TStrings;  
const Username, Password: string): Boolean;
```

Een eenvoudig mail programma kan nu snel in elkaar gestoken worden. Er zijn enkele configuratievariabelen nodig voor het afzender e-mail adres, de SMTP server, en gebruikersnaam en wachtwoord. Deze zijn als eenvoudige form-variabelen gedefinieerd:

```
FSender,  
FSMTPHost,  
FSMTPUser,  
FSMTPPasswd : String;
```

Ze kunnen via een configuratie dialoog ingevuld worden, en worden in een INI bestand bewaard en opgehaald bij het starten van het programma. Deze code is eenvoudig, en we bekijken ze dan ook niet in detail.

Om een mailte samen te stellen, zijn 2 `TEdit` controls voor de ontvangers en het onderwerp van de mail (respectievelijk `ETo` en `ESubject`), plus een `TMemo` (`MMail`) om het mail bericht in te geven. Een toolbar met een 'Verzenden' button vervolledigt het programma. De Verzenden knop is aan een actie gekoppeld, waarvan de `OnExecute` event handler als volgt invullen

```
procedure TMainForm.AVerzendenExecute(Sender: TObject);  
begin  
  If not CheckSettings then  
    Exit;  
  DoSendMail(ETo.Text, ESubject.Text, MMail.Lines);  
end;
```

De `CheckSettings` functie gaat na of er een afzender adres en SMTP server adres ingevuld zijn in de form-variabelen. Indien niet, wordt een boodschap getoond. Indien alles in orde is, wordt de functie `DoSendMail` opgeroepen. Die functie roept eenvoudigweg de `SendTo` of `SendToEx` functie op:

```
procedure TMainForm.DoSendMail(Const ATo,ASubject : String;
                               Content : TStrings);

Var
  B : Boolean;

begin
  if (FSMTPUser<>'') then
    B:=SendToEx(FSender,ATo,ASubject,FSMTPHost,Content,
               FSMTPUser,FSMTPPasswd)
  else
    B:=SendTo(FSender,ATo,ASubject,FSMTPHost,Content);
  if not B then
    ShowMessage('Kon het bericht niet verzenden!')
  else
    ShowMessage('Bericht verzonden.');
```

end;

Na afloop van de operatie, wordt een bevestiging getoond.

Meer is er niet nodig om een mail te sturen vanuit een programma.

3 Attachments

Vaak wil men een HTML mail sturen, of een bijlage met een logboodschap, of een PDF of beeldje. Het SMTP mail protocol staat alleen het sturen van platte tekst toe. Om een bijlage zoals een beeldje te versturen, moet MIME encoding toegepast worden. (MIME staat voor Multipurpose Internet Mail Extensions). MIME encoding zorgt ervoor dat een willekeurig aantal bestanden als tekst in het E-mail bericht kan gezet worden. Elk E-mail programma verstaat dit en kan de bijlages opnieuw samenstellen uit de MIME mail tekst.

Een MIME bericht bestaat uit verschillende delen (parts). Er zijn verschillende delen of Parts:

Text Bevat platte tekst (in een willekeurige karakterset).

Binary Bevat binaire data.

Message Bevat een compleet (ander) email bericht.

Multipart Bevat een of meerdere andere parts.

Om een mail te maken met een bijlage moeten we een 'multipart' part maken, waaraan dan een text part (voor de tekst van de mail) en een (of meerdere) binary parts worden toegevoegd. Het geheel wordt dan als mailbericht verzonden.

Synapse biedt een klasse aan om het samenstellen van een MIME email tekst te vereenvoudigen: `TMimeMess`. Deze klasse verzamelt enkele parts (een of meerdere instances van `TMimePart`), en stelt dan de MIME inhoud van het bericht samen. De klasse zorgt ook voor het verzamelen van de headers voor het mailbericht.

Een part wordt toegevoegd d.m.v. de `AddPart` functie:

```
function AddPart(const PartParent: TMimePart): TMimePart;
```

De `TMimePart` klasse heeft een plethora aan properties die de inhoud beschrijven. In principe moeten deze correct ingevuld worden opdat de `TMimeMess` instance het MIME bericht correct zou kunnen samenstellen.

Gelukkig zijn er wat hulp methodes die deze properties automatisch zetten. Een `MultiPart` deel toevoegen kan via:

```
function AddPartMultipart(const MultipartType: String;
                          const PartParent: TMimePart): TMimePart;
```

Tekst toevoegen kan via de volgende functies:

```
function AddPartText(const Value: TStrings;
                    const PartParent: TMimePart): TMimepart;
function AddPartTextEx(const Value: TStrings;
                      const PartParent: TMimePart;
                      PartCharset: TMimeChar; Raw: Boolean;
                      PartEncoding: TMimeEncoding): TMimepart;
function AddPartTextFromFile(const FileName: String;
                             const PartParent: TMimePart): TMimepart;
```

Binaire data kan aangehecht worden via:

```
function AddPartBinary(const Stream: TStream;
                      const FileName: string;
                      const PartParent: TMimePart): TMimepart;
function AddPartBinaryFromFile(const FileName: string;
                              const PartParent: TMimePart): TMimepart;
```

Alle functies hebben als laatste argument `PartParent`. Dit is het 'multipart' deel waaraan het deel moet worden toegevoegd: alleen multipart delen kunnen andere delen bevatten. Proberen een 'binary' part aan een ander binary part te koppelen zal een fout geven.

Er zijn nog wat hulpfuncties voor het toevoegen van HTML gegevens en het toevoegen van complete mail berichten. Eenmaal alle delen aan het bericht zijn toegevoegd, moet het bericht samengesteld worden (geencodeerd). Dit kan via de functie `EncodeMessage`.

Het gebruik van de `TMimeMess` klasse is eenvoudig te demonstreren: in de applicatie worden de bestandsnamen van de bijlages in een stringlist bijgehouden: (`FAttachments`).

De `OnExecute` handler van de verzenden actie wordt nu wat uitgebreider:

```
procedure TMainForm.AVerzendenExecute(Sender: TObject);
begin
  If not CheckSettings then
    Exit;
  if (FAttachments.Count>0) then
    DoSendMailAndAttachments(ETo.Text, ESubject.Text, MMail.Lines)
  else
    DoSendMail(ETo.Text, ESubject.Text, MMail.Lines);
end;
```

Als er bijlages zijn, wordt de functie `DoSendMailAndAttachments` opgeroepen, die het eigenlijke werk doet:

```

procedure TMainForm.DoSendMailAndAttachments (
    Const ATo,ASubject : String;
    Content : TStrings);

Var
    Mime : TMimeMess;
    P : TMimePart;
    I : Integer;
    B : Boolean;

begin
    Mime:=TMimeMess.Create;
    try
        // Zet enkele headers van het bericht.
        Mime.Header.ToList.Text:=ATo;
        Mime.Header.Subject:=ASubject;
        Mime.Header.From:=FSender;
        // Maak een multipart deel aan.
        P:=Mime.AddPartMultipart('mixed',Nil);
        // Als eerste deel de tekst van de mail toevoegen.
        Mime.AddPartText(Content,P);
        // Alle bijlagen toevoegen.
        For I:=0 to FAttachments.Count-1 do
            Mime.AddPartBinaryFromFile(FAttachments[I],P);

```

Na deze code is het MIME bericht klaar. Het bericht dat verzonden moet worden, kan nu samengesteld worden door EncodeMessage op te roepen, en met SendToRaw te versturen:

```

    // Bericht samenstellen
    Mime.EncodeMessage;
    // En verzenden met SendToRaw
    B:=SendToRaw(FSender,ATo,FSMTPHost,Mime.Lines,
                FSMTPUser,FSMTPPasswd);
    if not B then
        ShowMessage('Kon het bericht niet verzenden!')
    else
        ShowMessage('Bericht verzonden.');
```

```

finally
    Mime.Free;
end;
end;
end;
```

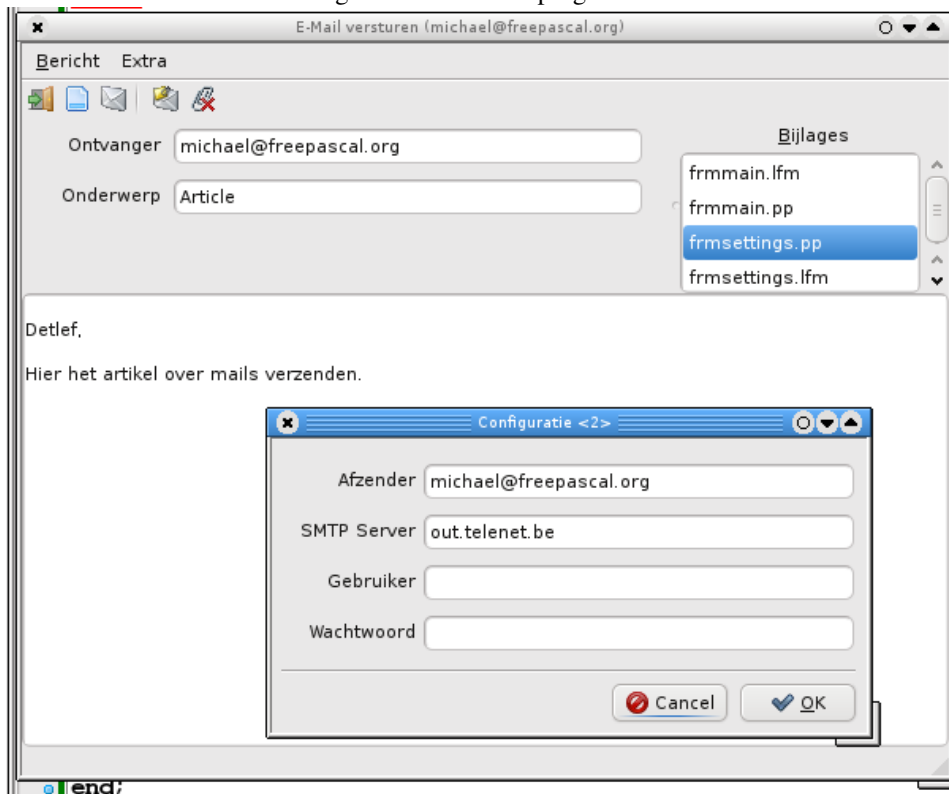
Nadat EncodeMessage is opgeroepen, bevat de property Lines van de TMimeMess klasse het geencodeerde bericht, inclusief mail headers. Dit bericht moet met SendToRaw verstuurd worden. De functie SendToRaw is als volgt gedefinieerd:

```

function SendToRaw(const MailFrom, MailTo, SMTPHost: string;
    const MailData: TStrings;
    const Username, Password: string): Boolean;
```

Deze functie verschilt van SendTo en SendToEx doordat het geen e-mail headers samenstelt; de E-mail headers moeten al in MailData aanwezig zijn. TMimeMess stelt deze headers samen in EncodeMessage.

Figure 1: Het e-mail programma



Het beheer van de bijlagen gebeurt via een listbox (LBAttachments). Een pop-up menu toont 2 menu items, 1 om een bijlage toe te voegen, 1 om de geselecteerde bijlage te verwijderen. Dezelfde acties kunnen via knopjes in de knoppenbalk uitgevoerd worden.

Het afgewerkte programma (met configuratiescherm) wordt getoond in figure 1 on page 6.

4 Verzendstatus

Zelfs in deze tijden van supersnel internet kan het versturen van grote bijlages nog flink lang duren. Gedurende die tijd is het programma bevroren, en heeft de gebruiker geen indicatie van de voortgang van het versturen van de E-Mail. Om dit te remediëren, heeft de TSMTPSend klasse een event die op gezette tijden opgeroepen wordt en waar de huidige toestand doorgegeven wordt. De event handler is van het volgende type:

```
THookSocketStatus = procedure(Sender: TObject;  
    Reason: THookSocketReason;  
    const Value: string) of object;
```

Sender is de TCP/IP socket waarvoor de status gerapporteerd wordt, Reason is een status indicator, die een van de volgende waarden kan aannemen voor een client socket:

HR_ResolvingBegin De server hostname wordt naar een IP adres omgezet.

HR_ResolvingEnd De server hostname is naar een IP adres omgezet.

HR_SocketCreate De TCP/IP socket voor communicatie met de server is aangemaakt.

HR_SocketClose De TCP/IP socket voor communicatie met de server is gesloten.

HR_Connect Er is verbinding gemaakt met de SMTP server.

HR_WriteCount Er is data pakketje naar de server gestuurd. Dit wordt vele keren gerapporteerd, en Value bevat het aantal verstuurde bytes in het pakketje.

HR_Error als er een fout optreedt.

De Value parameter bevat extra informatie over de status in een tekstuele vorm.

Om deze event te kunnen gebruiken, kunnen de eerder besproken functie SendTo, SendToEx en SendToRaw niet gebruikt worden. Het is nodig zelf een instance van de klasse TSMTPSend aan te maken, en de benodigde methodes op te roepen. Dit is niet veel werk, en de methode SendToRaw toont hoe de klasse te gebruiken:

```
Function TMainForm.DoSendMailAndAttachmentsProgress
  (Const ATo, ASubject : String; Content : TStrings) : Boolean;

var
  SMTP: TSMTPSend;
  s, t: string;
  L : Integer;

begin
  Result := False;
  SMTP:=TSMTPSend.Create;
  try
    SMTP.TargetHost := Trim(FSMTPHost);
    SMTP.Username := FSMTPUser;
    SMTP.Password := FSMTPPasswd;
    // Stel status callback in
    SMTP.Sock.OnStatus:=@ShowStatus;
    CurrentSent:=0;
    SendSize:=Length(Content.Text);
```

Hier zijn alle properties ingesteld die de TSMTPSend klasse nodig heeft om zijn werk te doen. De CurrentSent en SendSize zijn 2 form variabelen die het reeds verzonden aantal bytes en het totaal aantal te verzenden bytes bijhouden.

Nu kan het eigenlijke zenden beginnen:

```
// Aanmelden aan SMTP server
if SMTP.Login then
  begin
    // Stel From adres en totale berichtgrootte in
    if SMTP.MailFrom(GetEmailAddr(FSender), SendSize) then
      begin
        s:=ATo;
        // meld alle recipient adressen
        repeat
          t:=GetEmailAddr(Trim(FetchEx(s, ',', '"')));
          if (t<>'') then
            Result := SMTP.MailTo(t);
          if not Result then
```

```

        Break;
    until s = '';
    // verzend nu E-Mail inhoud
    if Result then
        Result := SMTP.MailData(Content);
    end;
    // En meld af.
    SMTP.Logout;
end;
finally
    SMTP.Free;
end;
end;

```

De `GetEmailAddr` uit de unit `synutil` haalt het email adres uit een adres-met-naam zoals

```
Michael Van Canneyt <michael@freepascal.org>
```

Rest nog de event handler te implementeren. Om de status aan te geven zetten we een progress bar en een status panel op de form. In een grote case statement worden de verschillende mogelijke status meldingen behandeld. De progress bar wordt pas getoond als er verbinding gemaakt is met de server, en wordt weer verborgen als de verbinding verbroken wordt.

```

procedure TMainForm.ShowStatus(Sender: TObject;
    Reason: THookSocketReason;
    const Value: string);

Var
    S : String;
    d : double;

begin
    Case reason of
        HR_ResolvingBegin :
            SBMain.SimpleText:='Opzoeken SMTP server IP';
        HR_ResolvingEnd :
            SBMain.SimpleText:='';
        HR_SocketCreate :
            begin
                PBStatus.Visible:=True;
                PBStatus.Position:=0;
            end;
        HR_Connect :
            SBMain.SimpleText:='Verbonden met SMTP server';
        HR_SocketClose :
            PBStatus.Visible:=False;
    end;
end;

```

De `HR_WriteCount` status wordt na elk verzonden pakketje opgestuurd. De volgende code houdt een totaal bij, en toont wat statistieken over de verzonden data in de status balk:

```

HR_WriteCount :
    begin

```



```
CurrentSent:=CurrentSent+StrToInt(Value);
D:=CurrentSent/SendSize;
PBStatus.Position:=Round(D*PBStatus.Max);
S:=Format('%d/%d bytes verzonden (%5.2f %%)',
          [CurrentSent, SendSize, D*100]);
SBMain.SimpleText:=S;
end;
end;
Application.ProcessMessages;
end;
```

Als laatste wordt `Application.ProcessMessages` opgeroepen zodat het scherm opnieuw getekend wordt.

5 Conclusie

In de on-line wereld van vandaag behoren taken zoals mails versturen tot de standaard functionaliteit van vele programmas. Lazarus biedt met synapse een eenvoudige manier om mails te versturen, zoals dit artikel hopelijk heeft duidelijk gemaakt.